

#2
500.36133X00

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant(s): T. SHIMOKAWA, et al
Serial No.:
Filed: March 19, 1998
Title: DATA STRUCTURE IN DATABASE, DATABASE SYSTEM
FOR MANAGING DATABASE AND DATABASE MANAGING
METHOD AND SYSTEM
Group:



LETTER CLAIMING RIGHT OF PRIORITY

Honorable Commissioner of
Patents and Trademarks
Washington, D.C. 20231

March 19, 1998

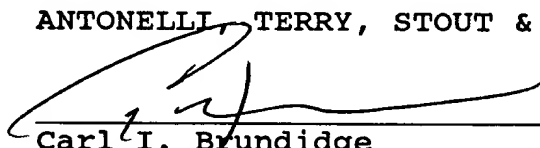
Sir:

Under the provisions of 35 USC 119 and 37 CFR 1.55, the applicant(s) hereby claim(s) the right of priority based on Japanese Patent Application No.(s) 9-065919 filed March 19, 1997.

A certified copy of said Japanese Application is attached.

Respectfully submitted,

ANTONELLI, TERRY, STOUT & KRAUS, LLP



Carl I. Brundidge
Registration No. 29,621

CIB/nac
Attachment

日 本 国 特 許 庁

PATENT OFFICE
JAPANESE GOVERNMENT

JCS30 U.S. PTO
09/044163
03/19/98

別紙添付の書類に記載されている事項は下記の出願書類に記載されて
る事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed
this Office.

願 年 月 日
Date of Application:

1997年 3月19日

願 番 号
Application Number:

平成 9年特許願第065919号

願 人
Applicant(s):

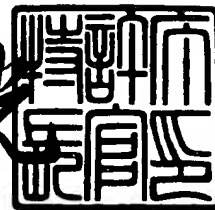
株式会社日立製作所

CERTIFIED COPY OF
PRIORITY DOCUMENT

1998年 1月 9日

特許庁長官
Commissioner,
Patent Office

荒井寿光



出証番号 出証特平09-3109630

【書類名】 特許願

【整理番号】 K97003891

【提出日】 平成 9年 3月19日

【あて先】 特許庁長官殿

【国際特許分類】 G06F 17/30

【発明の名称】 データベースのデータ構造及びデータベースのデータ処理方法

【請求項の数】 8

【発明者】

 【住所又は居所】 神奈川県横浜市戸塚区戸塚町5030番地 株式会社日立製作所 ソフトウェア開発本部内

 【氏名】 下川 隆義

【発明者】

 【住所又は居所】 神奈川県横浜市戸塚区戸塚町5030番地 株式会社日立製作所 ソフトウェア開発本部内

 【氏名】 正井 一夫

【特許出願人】

 【識別番号】 000005108

 【郵便番号】 101

 【住所又は居所】 東京都千代田区神田駿河台四丁目6番地

 【氏名又は名称】 株式会社日立製作所

 【代表者】 金井 務

【代理人】

 【識別番号】 100068504

 【郵便番号】 100

 【住所又は居所】 東京都千代田区丸の内一丁目5番1号 株式会社日立製作所内

 【弁理士】

 【氏名又は名称】 小川 勝男

【電話番号】 03-3212-1111

【手数料の表示】

【予納台帳番号】 013088

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【包括委任状番号】 9003094

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 データベースのデータ構造及びデータベースのデータ処理方法

【特許請求の範囲】

【請求項1】

データベースシステムに到着するデータを一定の時間分まとめ、連続するデータベースのエリアの管理単位セグメント群にまとめて格納し、当該セグメント群を管理するブックマークとして時間の管理情報を格納する領域を持つことを特徴とするデータベースのデータ構造。

【請求項2】

データベースに到着するデータを一定の時間分まとめ、連続するデータベースのエリア管理単位セグメント群にまとめて格納し、当該セグメント群を管理する管理情報を格納するまでデータベースとしては挿入されていることをエリア的に存在しない管理をして無効とすることを特徴とするデータベースのデータ処理方法。

【請求項3】

データベースに到着するデータを一定の時間分まとめ、連続するデータベースのエリア管理単位セグメント群にまとめて格納し、データを削除する際に、削除すべきセグメント群をブックマークから検出し、該セグメントエリアを空セグメントエリアに管理情報上で変更することを特徴とするデータベースのデータ処理方法。

【請求項4】

データベースに到着するデータを一定の時間分まとめ、連続するデータベースのエリア管理単位セグメント群にまとめて格納し、当該セグメント群を管理するデータ格納時間情報を格納する領域を有し、データを削除する際には、削除すべき前記セグメント群を前記データ格納時間情報から検出し、当該セグメントエリアを空セグメントエリアに前記データ格納時間情報に変更することを特徴とするデータベースのデータ処理方法。

【請求項5】

データベースを有するコンピュータシステムと時系列順にデータが到着し、時

系列のデータ挿入及び削除以外のデータ変更が無い検索用データベースにおいて到着するデータを一定の時間分まとめて連続するデータベースのエリアの管理単位セグメント群にまとめて格納し、それらセグメント群を管理するブックマークとして時間の管理情報を格納する領域を持つことで、索引を作成することなく、時間を特定した検索において、データベース全体を検索しなくてもその該当セグメントだけを検索することで検索を高速化することを特徴とする時系列ブックマークを用いる時系列データベースの管理方式。

【請求項6】

データベースを有するコンピュータシステムと時系列順にデータが到着し、時系列のデータ挿入及び削除以外のデータ変更がない検索用データベースにおいて到着するデータを一定の時間分まとめて連続するデータベースのエリア管理単位セグメント群にまとめて格納するが、それらセグメント群を管理するブックマークとして時間の管理情報を格納するまでデータベースとしては挿入されていることをエリア的に存在しない管理をして有効としないことでオンライン中にデータロードを可能とすることを特徴とする時系列データベースのデータ追加方式。

【請求項7】

データベースを有するコンピュータシステムと時系列順にデータが到着し、時系列のデータ挿入及び削除以外のデータ変更がない検索用データベースにおいて到着するデータを一定の時間分まとめて連続するデータベースのエリア管理単位セグメント群にまとめて格納するが、データ保存期間を越えたデータを削除する際には、削除すべきセグメント群をブックマークから検出し、該セグメントエリアを空セグメントエリアに管理情報上で変更することで内部のデータをアクセスすることなく短時間でかつオンライン中に削除を可能とすることを特徴とする時系列データベースのデータ削除方式。

【請求項8】

データベースを有するコンピュータシステムと時系列順にデータが到着し、時系列のデータ挿入及び削除以外のデータ変更がない検索用データベースにおいてデータベースに到着するデータを一定の時間分まとめて連続するデータベースのエリア管理単位セグメント群にまとめて格納するが、それらセグメント群を管理

するブックマークとして時間の管理情報を格納する領域を持つことで、データ保存期間を越えたデータを削除する際には、削除すべきセグメント群をブックマークから検出し、該セグメントエリアを空セグメントエリアに管理情報上で変更することで常に連続エリアを確保し、セグメントエリアをラップアラウンド的にしようとすることで追加削除を繰り返すが、いっさい再編成を必要としないことをことを特徴とする時系列データベースのデータエリア管理方式。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、データの更新を直接行わずに、時系列順に更新明細情報として格納し、時系列順の分析を行うことを目的とする特に超大規模な時系列データベース処理方式に関する。

【0002】

【従来の技術】

大規模なデータベースから特定のデータを検索する場合、索引（インデクス）をつけることが一般的である。インデクスは、検索をする際のキーとなる項目が特定できる場合には有効である。インデクスは、データベースの特定のキー項目を集めて、その上位にバランストリーの形にポインタを設け、特定の値のキーがどの範囲にあるかという情報からトリーのリーフに当たる位置まで高速にたどれるようにした仕掛けであり、アン イントロダクション トゥ データベース システムズ、シー ジェイ デイト、アディソン ウェズリー、1986、第68頁から第77頁（An Introduction to Database systems、C. J. Date、ADDISON-WESLEY、1986、pp68-pp77）において論じられている様に、全データ項目についてその格納位置に相当する情報が得られる仕掛けになっている。

【0003】

しかし、百万件程度のデータベースでは問題がないが、十億件とか一兆件といった超大規模なデータベースにおいては、インデクスそのものの維持が膨大となり、特に時系列で追加をしていくようなキーに関しては不適切である。

【0004】

時系列でデータを追加すると、時間を示す項目の値は単純増加をするため、インデクスは一定の方向に成長する。さらに削除という観点では、一定時間経過したものを削除していくとインデクスは片方の部分にばかりデータがあり、手前の方のノードが存在するのに項目の値がない形となり、非常に効率の悪い状態となることが知られている。この際は、再編成という技術でインデクスを再作成してインデクス内の無駄エリアを削除し、効率を上げるということをする。これも、超大規模な時系列データベースでは許容範囲をはるかに超える作業となり現実的でない。

【0005】

データベースに高速にデータを追加するには、データロード用のユティリティを用いる。データロードユティリティを用いるとデータベースの物理的なエリアに直接データを書き込む技術を用いているため高速に書き込むことが可能である。しかし、物理的なエリアに直接書き込む高速なデータロード用のユティリティは、その特性上、他の検索や更新アクセスと競合することを禁止している。すなわち、その特定の表または、表の一部のアクセスを禁止した状態で実行することが義務付けられている。これでは、毎日のように追加される時系列データをデータロードする度にデータベースの検索をいったん打ち切る必要がある。超大規模な時系列データベースにおいては、検索そのものが一日以上かかるケースが有る。この場合、検索を打ち切らないとデータロードできないことが非常に大きな弊害をもたらす。この状態を避けるために、データロードではなく、通常データ挿入操作でデータを追加することも可能だが、物理的に書き込むタイプのデータローダに比べて一桁近く性能が悪くなる上に、追加中のデータを見せなくするためのロックを取得する必要がある、データベース全件を検索している操作に対する性能的影響が大きい。

【0006】

また、データベースの中の一定期間を経過したデータを削除する場合、通常は、該当するデータを検索し、一件ずつ削除していく。このため、インデクスがあっても該当する不要データを削除するためには、データを一件ずつ挿入するとき

に匹敵する時間がかかる。もしインデックスを持たない場合には、全件を検索し該当データを削除するため、超大規模なデータベースにおいては、削除処理だけで一日以上かかる操作となり、時系列データベースとして現実的に成立しない。

【0007】

一般に追加削除を繰り返すとデータベースの中は乱れるため、再編成を定期的に行う必要がある。この再編成は、全データを読み出し、格納し直すために、非常に時間のかかる操作である。超大規模データベースでは、再編成をすることは、数日以上かかる仕事となり現実的ではない。

【0008】

【発明が解決しようとする課題】

上記従来技術は、データの検索の高速化のためには、キー項目一件ずつに対応するインデックスを作成していたが、超大規模データベースではインデックスのデータ量が大きくなりすぎてしまう。また、バランスドトリーの形では、時系列追加と削除を繰り返す操作に弱く、再編成を強いられる。この再編成の操作は、超大規模なデータベースにとって現実的ではない。

【0009】

データ追加に関しても、インデックスが存在するとそのインデックスの更新を伴い非常に時間がかかる。インデックスを追加しないと検索が全件を対象とするため、この場合も時間を非常に必要とする。さらに高速にデータ追加をすることを優先すると、他のデータ検索操作をいったん打ち切る必要がある。これでは、全件検索が一日以上かかるデータベースに対して毎日のデータ追加をすることは、現実的に実現困難である。

【0010】

さらに、一定時間を経過したデータの削除に関しても、インデックスがなければ全件検索時間以上にかかり、インデックスが存在すれば削除の際にインデックスの更新も伴い、データ挿入同様非常に時間のかかる操作となってしまう。これでは、全件検索が一日以上かかるデータベースに対して毎日のデータ削除をすることは、現実的に実現困難である。

【0011】

データを追加及び削除を繰り返す時系列データベースにおいては、データベースの乱れを直すために再編成を行うことになるが、これも超大規模なシステムにおいては数日以上かかる操作となってしまう現実的ではない。

【0012】

【課題を解決するための手段】

前記課題を解決するために本発明は、データベースのエリア管理の最小単位であるセグメント毎に分け、時系列データを格納する。データベースにデータを追加するときには、追加を開始する先頭のセグメントに対し、ブックマークとしてその追加をした時点の時刻を記憶する。ブックマークがあるので、時刻指定、または時間間隔指定の検索の場合、このブックマークを利用して検索する範囲を物理的に狭めることができる。

【0013】

データ追加を行う場合には、ブックマークを追加中であるという状態にすることでデータベースとしては追加が完了していない状態を作ることができる。このため、他の検索に影響を出すことなく物理的なセグメントに直接データロードを行うことができる。データ追加が完了した時点で、データベースとしてブックマークを付けて認知する。

【0014】

データ削除時には、特定のブックマーク以降のデータをまとめて削除する場合、セグメント単位に空エリア化することで、実際のデータをアクセスすることなく短時間で削除することができる。データベースのエリアをラップアラウンド型にセグメントという単位で管理することで、常にプールされている連続エリアを一方から使いデータを挿入し、その他方に対してデータ削除したエリアを補充することができる。

【0015】

また本発明は、データベースを有するコンピュータシステムと時系列順にデータがデータベースシステムに到着し、時系列のデータ挿入及び削除以外のデータ変更がない検索用データベースシステムに対して特に有効である。

【0016】

【発明の実施の形態】

以下、本発明に一実施例について図面を用いて説明する。

【0017】

図1は、本発明の一実施例を示すシステム構成図である。

データベースシステムは図1に示されるように、主として、中央処理装置11を有するシステム装置10とデータを物理的に格納する記憶装置13から構成される。データベース管理プログラム12はシステム装置10上で動作し、記憶装置13上に実際のデータを格納する。記憶装置13には、データ領域14とそのデータの定義情報を格納するためのシステム定義情報領域15が存在する。

【0018】

システム定義情報領域15とデータ領域14を更に詳細に説明したものが図2の記憶装置の構成図である。本実施例におけるデータ領域14は記憶装置13上の連続した領域に確保され、セグメント20という管理単位に分割される。時系列的に発生したデータは一定の時間分まとめて、連続する記憶装置13からなるデータベースのデータ領域14としてのデータが記憶可能なエリアの管理単位セグメント群としてまとめて格納される。そのセグメント20は、実データを格納する為のデータ格納領域21とそのデータ格納領域21に格納されているデータの管理情報を格納するためのブックマーク情報領域22とから構成される。本実施例では、セグメント20はディスク入出力の単位であるページが複数個で構成されている。

【0019】

システム定義情報15には、時系列データの格納位置を管理する情報が含まれており、最も時系列的に古いセグメント20をポイントする情報と、空セグメント領域の先頭をポイントする情報が含まれている。

【0020】

図3のブックマーク情報領域22の説明図で示すとおり、ブックマーク情報領域22では、セグメント20にデータが格納された時間に関する情報を時間情報領域23に、セグメント20の遷移状態を示すステータスフラグ情報をステータスフラグ領域24に格納する。ステータスフラグ情報としては、遷移状態にはそのデータ格納

領域がアクセス可能であることを表す「オンライン」。現在データを挿入中であることを表す「書き込み中」。そしてデータ格納領域にはデータが入っていないことを表す「空き」の三種類があり、セグメント20のステータスとしてはこれらの間を遷移する。

【0021】

次に、本実施例の動作について説明する。

時系列データベースにおいては、時間を特定した検索が頻繁に実施される。例えば、ある出版社の時系列データベースにその出版社から出版された本のタイトルと発行日が、発行日をキーとして時系列に格納されているような場合、このデータベースに対し1994年の3月から1994年の5月までの3カ月間に発行された本のタイトル一覧の検索などが一例となる。

【0022】

図4及び図5を用いて本実施例の検索処理を説明する。図4は本実施例の検索処理の一実施例を示す流れ図であり、図5は図4の流れ図を説明するための記憶装置の構成図である。

【0023】

本実施例のデータベースシステムはまず最も時系列的に古いデータが格納されているセグメント20をポイントする情報をシステム定義情報15から取得する（ステップ500）。データベースシステムは、ポイントとされているセグメント20のブックマーク情報領域22から、時間情報 t （1994年2月）とステータス情報（オンライン）を取得する（ステップ501）。

【0024】

取得したステータス情報が「空き」もしくは「書き込み中」であれば、そのセグメント20には対象となるデータが格納されていないかデータを挿入中であるためアクセスは不可と判断できるため、検索処理を終了する（ステップ502）。

【0025】

ステータス情報が「オンライン」であればアクセス可能となり次の処理（ステップ503）へ進む。時間の新しい方の検索要求時間（1994年5月）とブックマーク情報領域22に格納されていた時間情報（1994年2月）とを比較し、

このデータベースに目的のデータが格納されているかどうかを判定する。比較結果が「Yes」であった場合、格納されているデータは検索対象である範囲（1994年3月～1994年5月）よりも時間的に新しいデータが格納されているということなので、検索処理を終了する。判定処理が「No」である場合、次の処理（ステップ504）へ進み、現在ポイントしているセグメント20が検索要求期間内（1994年3月～1994年5月）であるかどうかを判定する。セグメント20は1994年2月であるため、このデータ格納領域21は検索対象外となり、時系列的に次のデータが格納されているセグメント20をポイントする（ステップ506）。例えば、セグメント20のサイズは全て同じサイズであるとする、記憶装置13として磁気ディスク装置を用いた場合、現在のディスクのヘッダからセグメント分のサイズ（磁気ヘッドに対する相対的な移動量）を動かすことにより、次のセグメントをポイントすることが可能になる。

【0026】

次セグメント20についても上記と同様（ステップ502、503、及び504）の判定処理を実行する。処理（ステップ504）においてセグメント20は検索要求に合致するセグメントであることが判定されると、セグメント20内のデータ格納領域21から実データを読み込む（ステップ505）。データ格納領域21から実データを読み込んだ後はディスクのヘッダは時系列的に次のデータが格納されているセグメント20の先頭を指していることになるので（ステップ506）、またブックマーク情報領域22から時間情報を取得し、以下、同様に判定を繰り返す。こうして、セグメント20が順々に読み出されることになる。セグメント20の判定処理（ステップ503）において、検索対象外であることが判定されるので、ここで検索処理は終了する。

【0027】

次にデータロード処理について図6を用いて説明する。図6は本実施例におけるデータロード処理を示す流れ図である。本実施例では、1994年7月分及び1994年8月分のデータがデータロードされる場合を説明する。入力データはファイルとして与えられ、データは既に時系列的にソートされているとする。図5で表される初期状態にデータが追加される例で説明する。

【0028】

まず、システム定義情報15から空セグメント情報を読み出す（ステップ600）。その情報により空セグメント20をポイントする。入力データを読み込むため入力ファイルにアクセスし、データ（1994年7月）を読み込む（ステップ602）。データが存在するため、処理（ステップ602）では「有」が判定され、処理（ステップ603）へ進む。処理（ステップ603）で書き込み処理が実行される。まず、ブックマーク情報領域22に時間情報（1994年7月）と現在ローディング中であることを表すフラグ「書き込み中」をステータスフラグ領域に書き込み、データ格納領域21に実データを書き込む。書き込み終わった状態を表したのが図7である。

【0029】

1つのセグメント分のデータが書き終わると、データベースシステムは次の入力データをファイルから読み込み（ステップ601）、1994年8月分のデータが存在するため、判定処理（ステップ602）で「有」の判定をする。1994年7月分のデータを書き込んだ時と同じロジックによりセグメント20に時間情報（1994年8月）、ステータスフラグ「書き込み中」、及び実データをデータ格納領域21に書き込む（ステップ603）。

【0030】

書き込みが終了すると、さらにシステムは次のデータをファイルから読み込もうとする（ステップ601）。しかしながらファイルには既にデータは存在しないため、判定処理（ステップ602）で「無」と判定され、次の処理（ステップ604）へ移行する。

【0031】

入力データのデータベースへの書き込みが終了すると、データベースシステムは新たなデータが書き込まれたセグメント群をアクセス可能にするため、ブックマーク情報領域のステータスフラグの更新を開始する（ステップ604）。

【0032】

最後のデータを書き込み終えたとき、データベースシステムはシステム定義情報15の空セグメント情報16を読み込み、新たにデータを最初に書き込んだセグメ

ント20をポイントする。そのブックマーク領域22のステータスフラグには「書き込み中」と設定されているので、このフラグを「オンライン」に遷移させる。これによりセグメント20に対しての検索が可能となる。本実施例において、検索の時と同様にセグメントのサイズは決められているため、時系列的に次の情報が格納されているセグメント20をポイントできる。

【0033】

セグメント20のブックマーク情報領域22のステータスフラグにも「書き込み中」が設定されているので、これを「オンライン」に設定し、セグメント20を検索が有効である状態へと遷移させる。この「書き込み中」から「オンライン」への遷移移行処理は、読み込んだブックマーク情報領域のステータスフラグが「空き」になったところで終了し、そのセグメントへのアドレス情報をシステム定義情報15の空セグメント情報16に設定する（ステップ605）。データロード処理が全て完了した状態が図8になる。

【0034】

次に削除処理について図9を用いて説明する。図9は削除処理の一実施例を示す流れ図である。

【0035】

本実施例では図5を初期状態とし、1994年2月分のセグメント20を削除する処理について説明する。

【0036】

まず、システム定義情報15から先頭セグメント情報20を読み出す（ステップ700）。セグメント20のブックマーク情報領域22から時間情報（1994年2月）を取得し、セグメント20が削除対象のセグメントであるかどうかを判定する（ステップ701）。

【0037】

削除対象は1994年2月であるので、そのセグメント20が削除対象と判定される。システム定義情報15の先頭セグメント情報16を時系列的に次のセグメント20へ変更する。セグメントサイズは決まっているため、先頭セグメントアドレスはセグメントのサイズ分だけ加算すればよい（ステップ702）。

【0038】

次に、ブックマーク情報領域22に時間情報（ナル値）を設定し（ステップ703）、ステータスフラグに「空き」を設定する（ステップ704）。ブックマーク情報領域22を初期化することにより（ステップ703及び704）、セグメント20をアクセス不可の状態へ遷移させる。

【0039】

時系列的に次のセグメント20をポイントし（ステップ705）、セグメント20からブックマーク情報領域22から時間情報（1994年3月）を取得する。取得した時間情報（1994年3月）と削除対象である1994年2月とを比較し、セグメント20が削除対象ではないこと判定する（ステップ701）。これで削除処理が終了する。終了後のデータベースの状態は図10のようになる。

【0040】

本削除処理は内部のデータを直接アクセスすることなく、ブックマーク情報領域のみを対象にすることで、短時間にかつオンライン中に削除が可能となる。

【0041】

また、セグメント群をラップアラウンド的に使用することで、追加削除を繰り返しても、一切再編成を必要としない特徴がある。最後にこのラップアラウンド機構について説明する。

【0042】

図11にラップアラウンド機構の実施例を示す。物理的に一番下位に位置するセグメント20のブックマーク情報領域22のステータスフラグ領域に「先頭」というフラグを設定し、ラップアラウンドを実現する。このステータスフラグ領域に「先頭」というフラグが設定されている場合、時間情報領域として時間情報を管理していた領域にセグメント群の先頭セグメント20のアドレスを設定しておく。検索・削除・挿入等の処理においても、常にブックマーク情報領域を参照する処理があるので、この処理の延長で先頭アドレスへジャンプする処理を加えるだけで容易に実現可能である。

【0043】

また、図12において示す本発明の他の実施例のように、ある一定期間のデー

タは複数のセグメント群20によって管理されている場合の方が多い。従って、本実施例の変形として、ブックマーク情報をシステム定義情報15内に格納し集中管理する方式がある。この方式はデータを書き込むべきエリアが限定できるというメリットがありより現実的である。

【0044】

以上述べたとおり、本発明の実施例を実施することによって、インデクスが無くともデータベース全体を検索すること無く特定の範囲だけを全件検索することで、目的とするデータをアクセスすることができる。

【0045】

また本発明の実施例を実施することによって、データ追加時には、データロードを事前に空きエリア上に行い、そのデータロードが完了した時点でブックマークを付けてデータベースの表として実際の追加をすることで他の検索を止めること無く同時にかつ非常に高速にデータロードができる。

【0046】

さらに本発明の実施例を実施することにより、一定の期間を超過したデータの削除に関しても、ブックマークを検索することで削除対象セグメントと特定でき、セグメントがデータベースのエリア管理の単位であるために空エリア化する事ができ、非常に短時間で（通常数秒から数分程度で）削除できる。

【0047】

また本発明の実施例で説明したように、本発明によるとエリア管理もラップアラウンド型にできるため、再編成は不要となる。

【0048】

本発明のによれば、ブックマーク情報は膨大な実データ量に比べその規模は小さくメンテナンス処理が非常に容易に実現でき、大規模データベースにおいても非常に短時間でブックマーク情報を検索することができる。

【0049】

【発明の効果】

本発明によれば、膨大な実データ量を有し、時系列順に到着するデータの記憶及び削除を行う大規模データベースにおいて、高速に検索しオンライン中であっ

てもデータロード・削除処理を実現することができる。

【図面の簡単な説明】

【図1】

本発明を実現する一実施例を示すシステム構成図である。

【図2】

本発明を実現する一実施例を示す記憶装置の構成図である。

【図3】

ブックマーク情報領域の説明図である。

【図4】

検索処理の一実施例を示す流れ図である。

【図5】

図4の流れ図を説明する為の記憶装置の構成図である。

【図6】

データロード処理の一実施例を示す流れ図である。

【図7】

図6の流れ図を説明する為の記憶装置の途中経過を表す構成図である。

【図8】

図6の流れ図を説明する為の記憶装置の最終結果を表す構成図である。

【図9】

削除処理の一実施例を示す流れ図

【図10】

図9の流れ図を説明する為の記憶装置の途中経過を表す構成図である。

【図11】

ラップアラウンド機構の一実施例を示す記憶装置の構成図である。

【図12】

本発明を実現する他の実施例を示す記憶装置の構成図である。

【符号の説明】

10…システム装置

11…中央処理装置

12…データベース管理プログラム

13…記憶装置

14…データ領域

15…システム定義情報領域

16…格納位置管理情報

20…セグメント

21…データ格納領域

22…ブックマーク情報領域

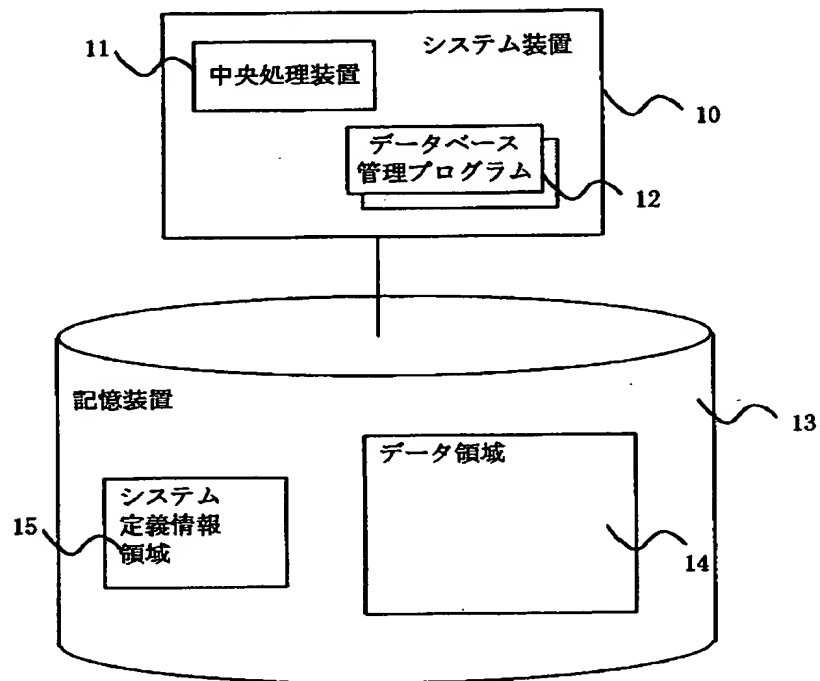
23…時間情報領域

24…ステータスフラグ領域

【書類名】 図面

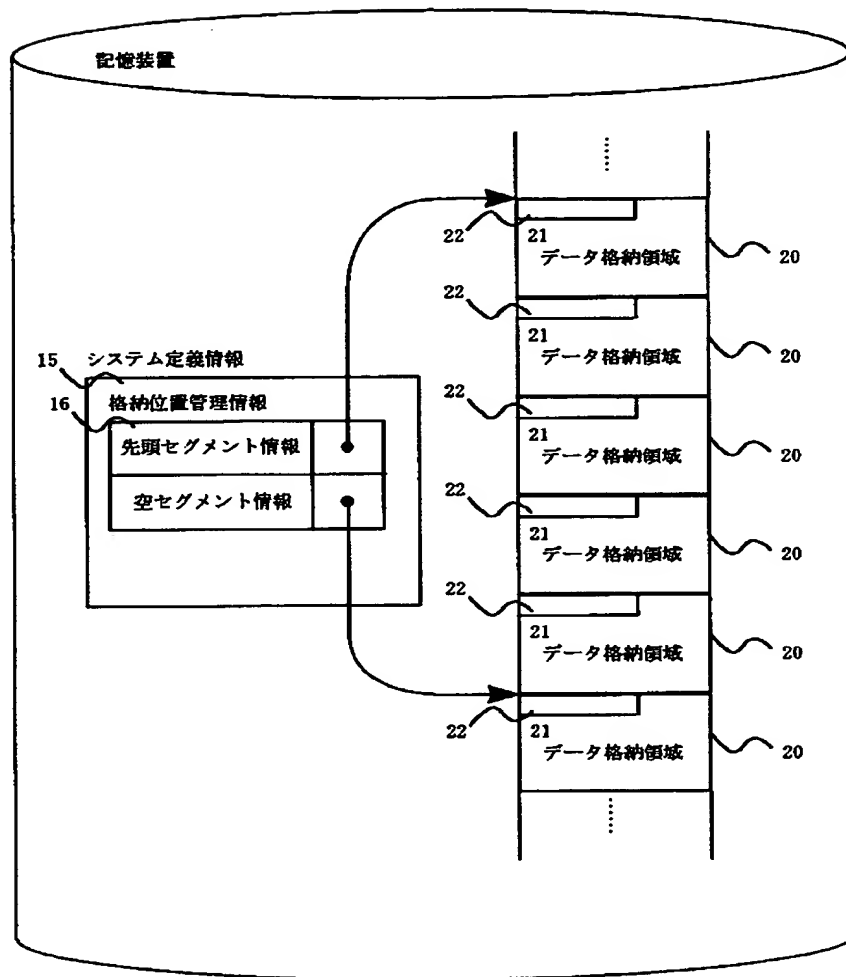
【図1】

図1

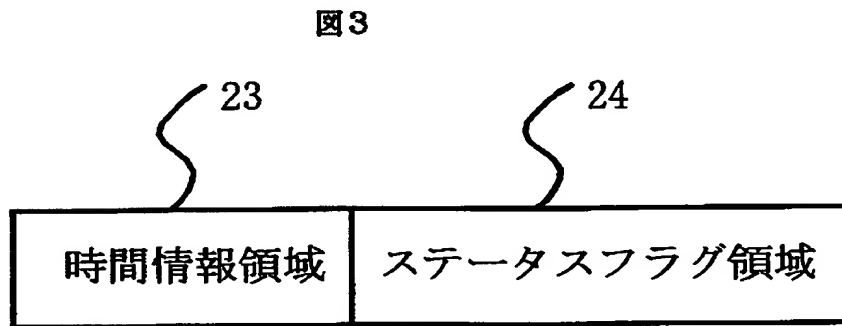


【図2】

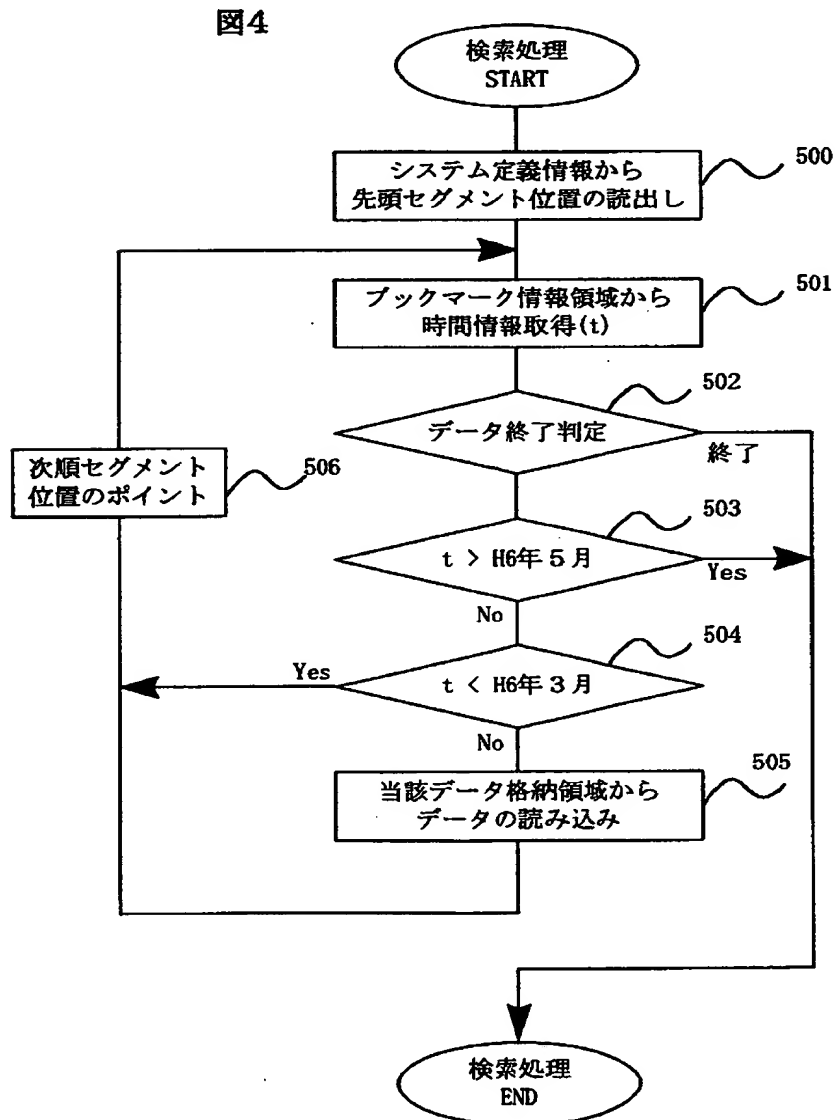
図2



【図3】

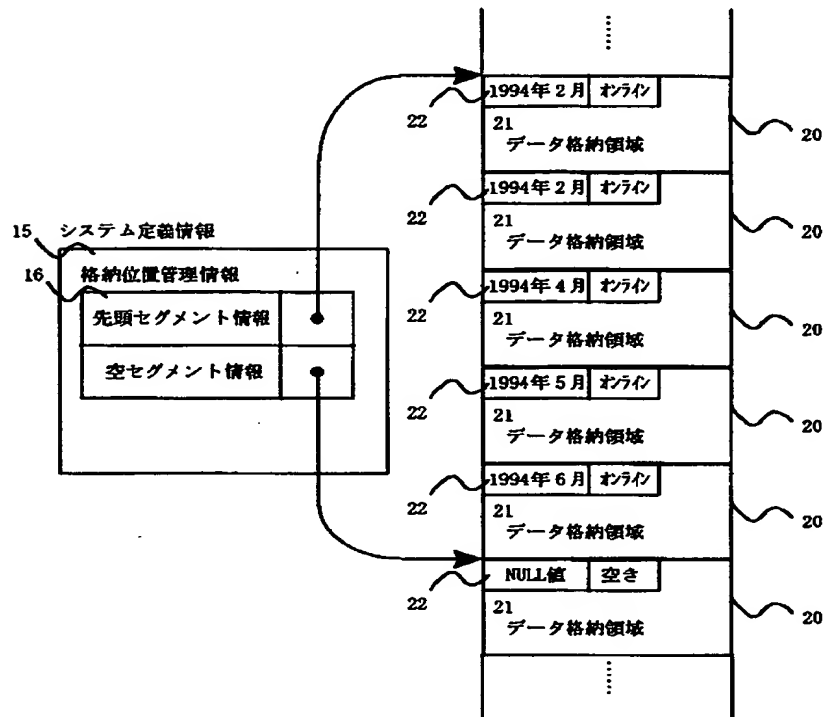


【図4】



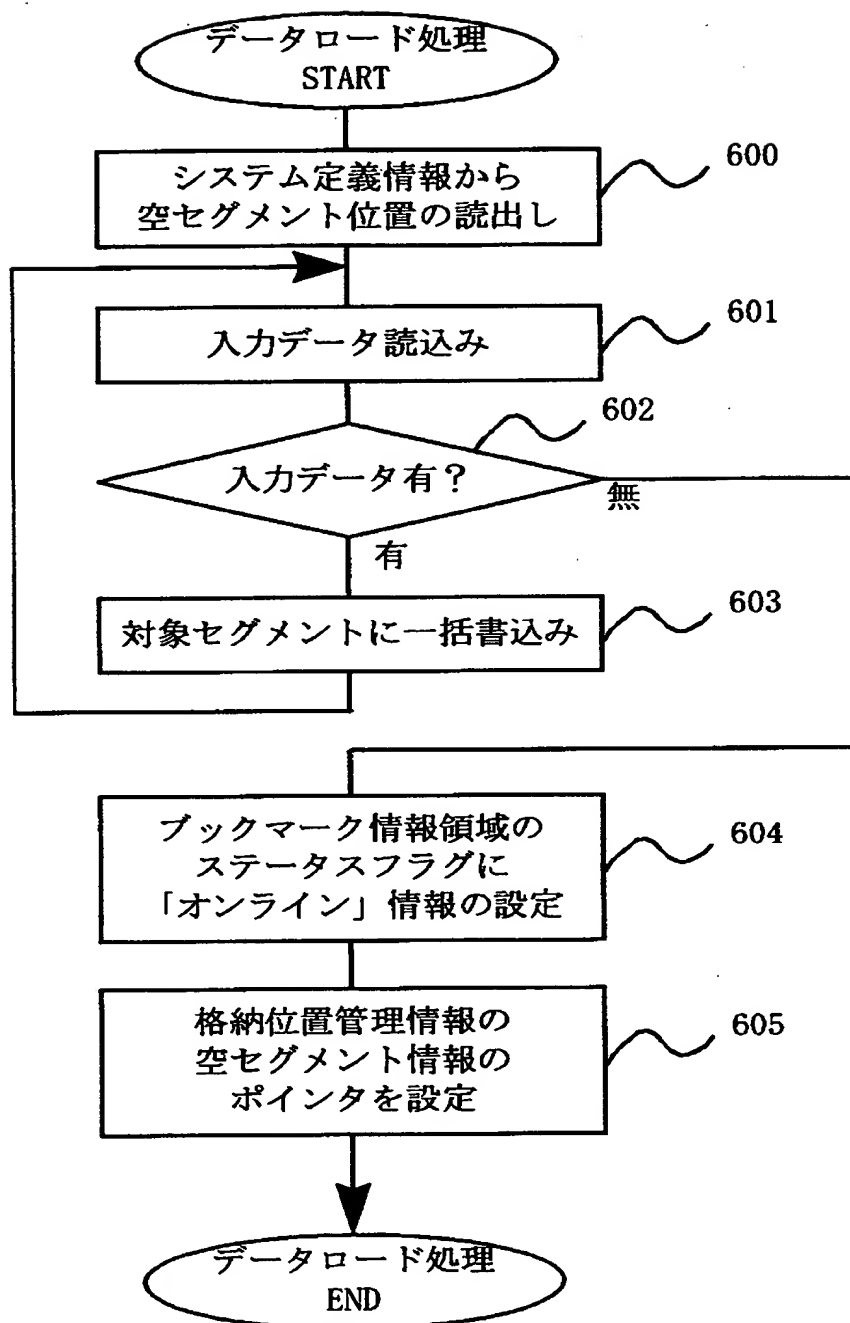
【図5】

図5



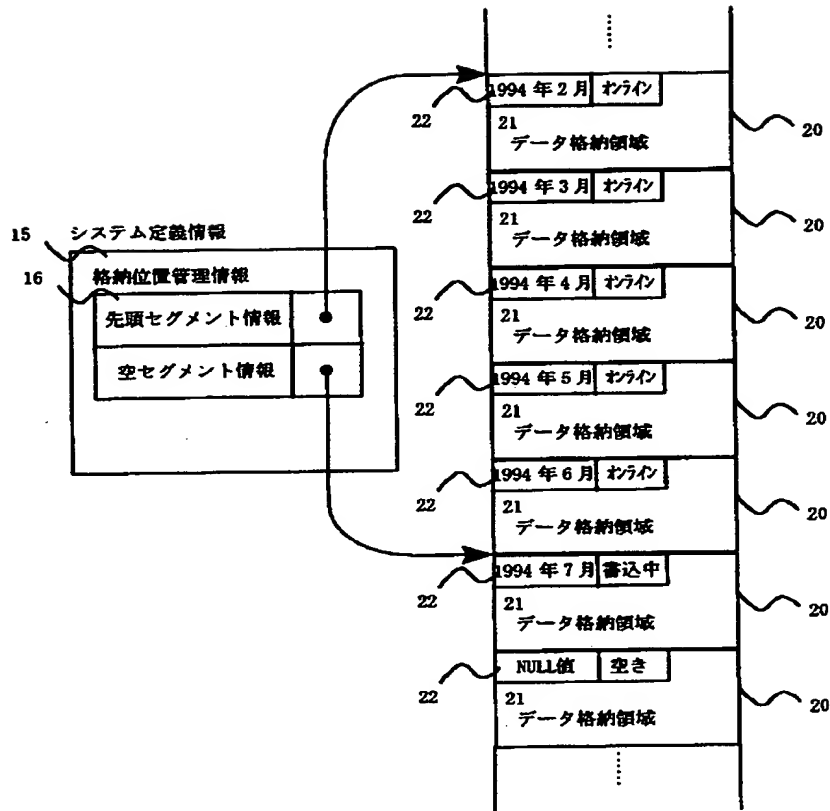
【図6】

図6



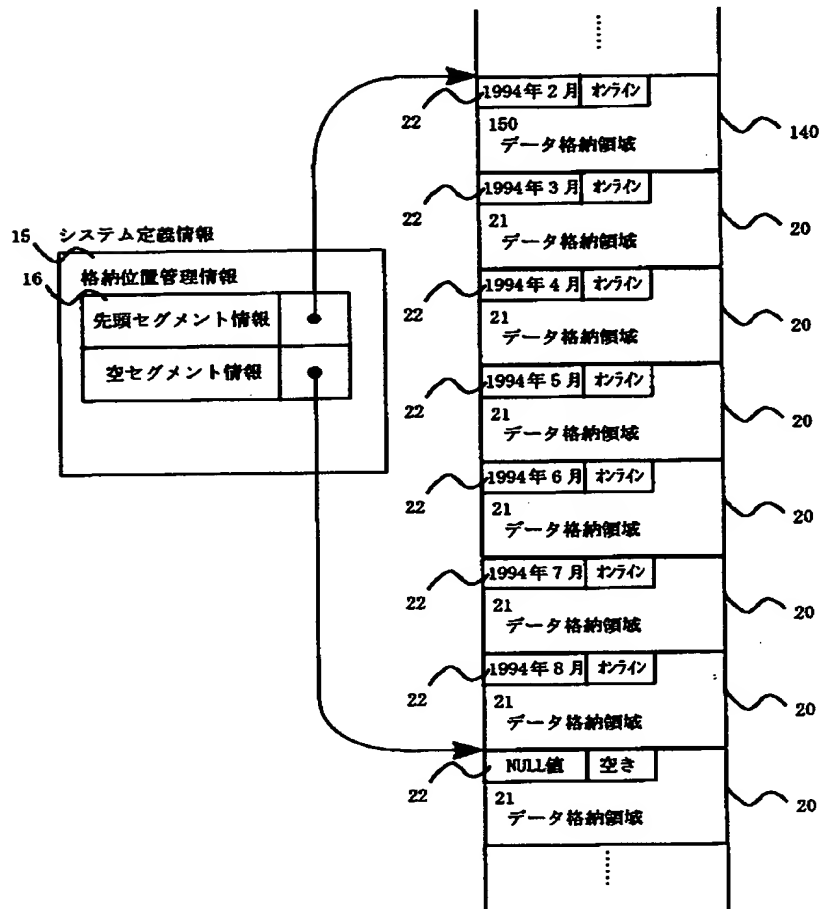
【図7】

図7



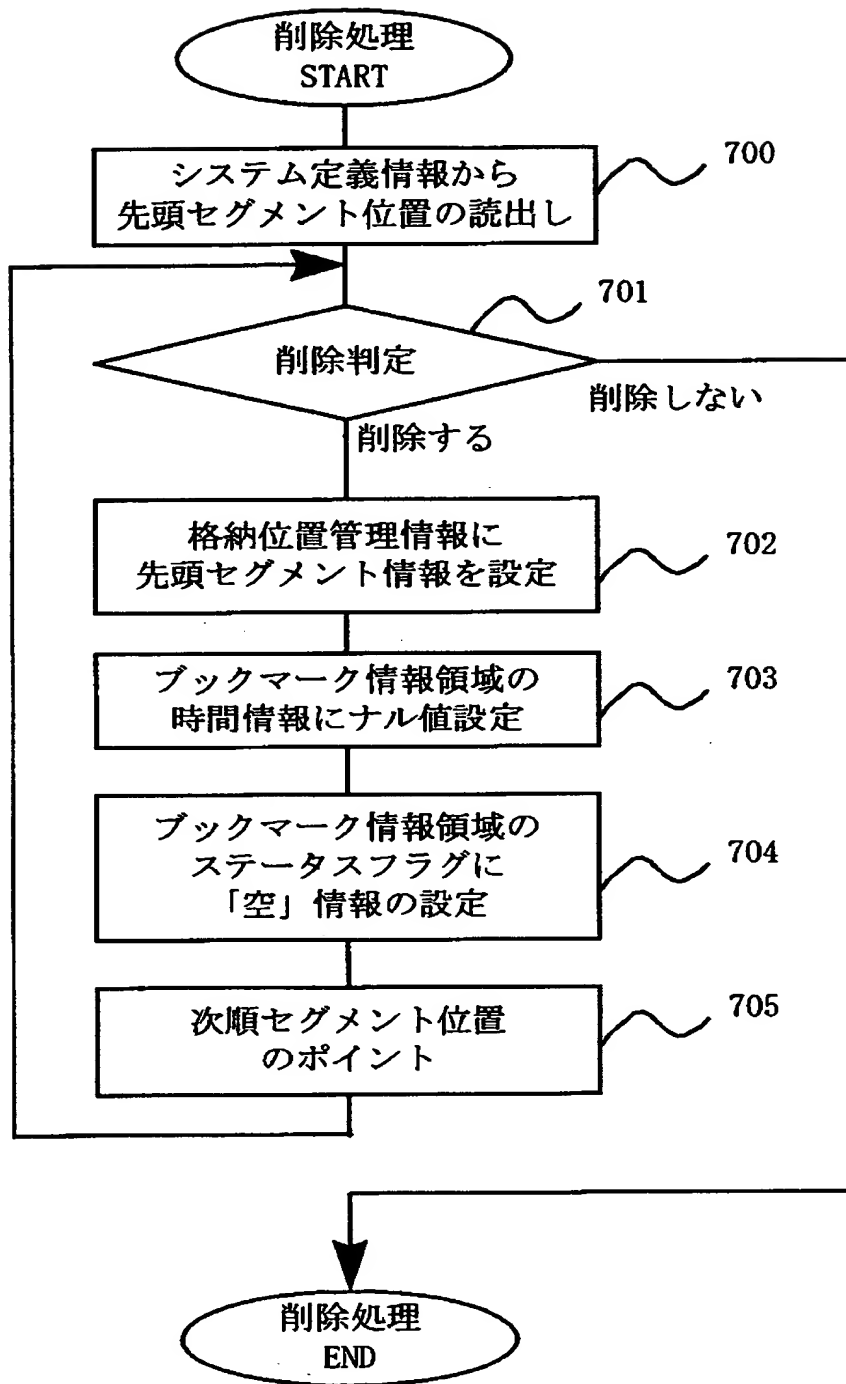
【図8】

図8



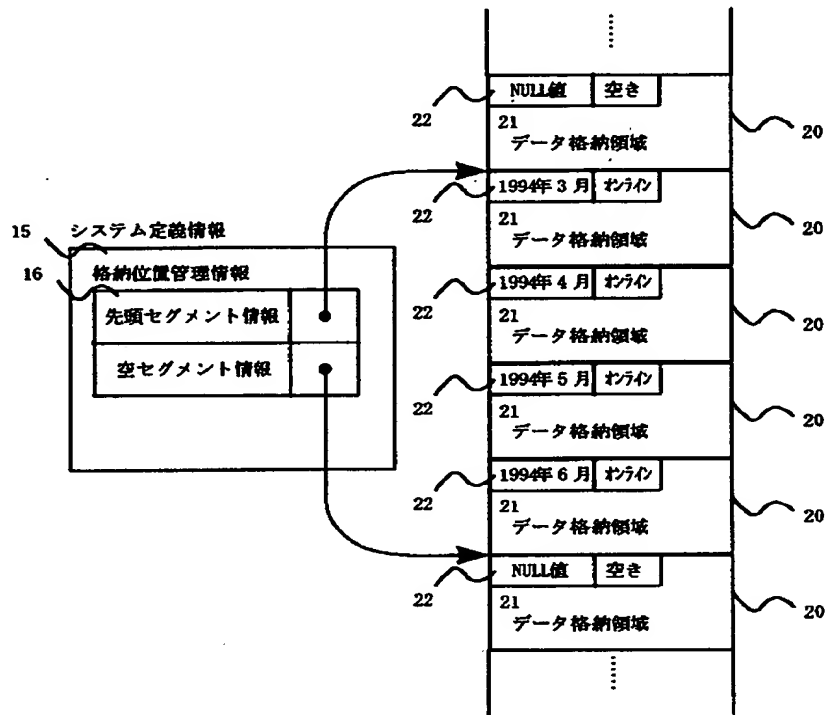
【図9】

図9



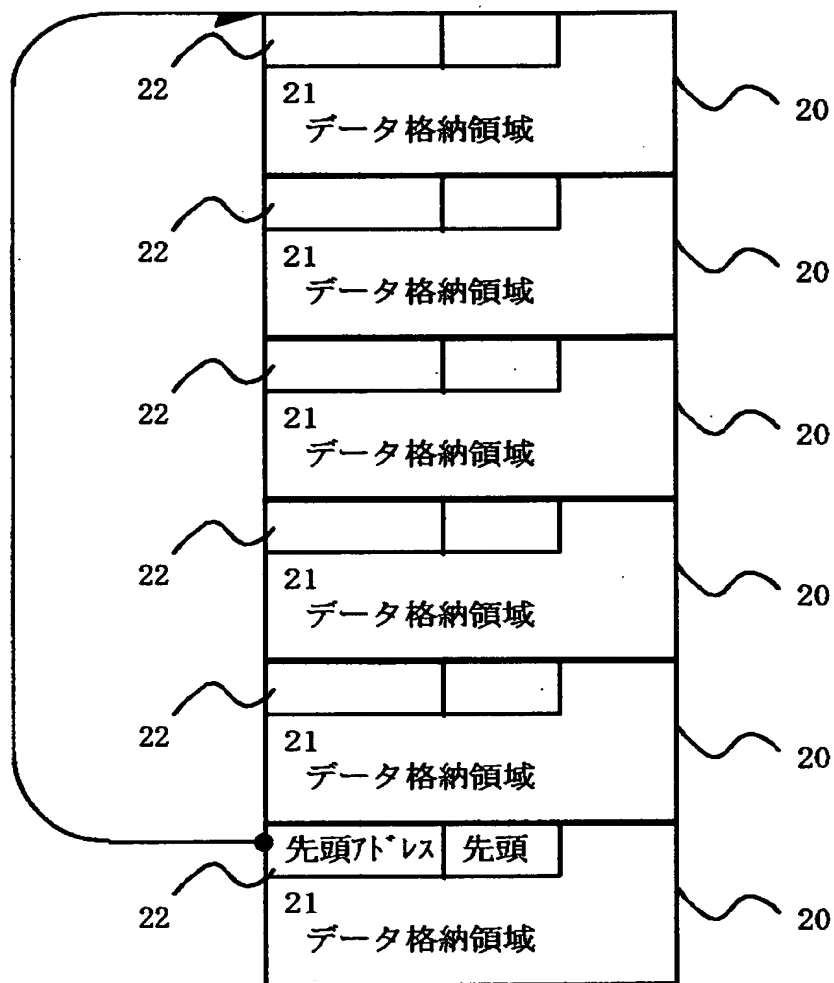
【図10】

図10



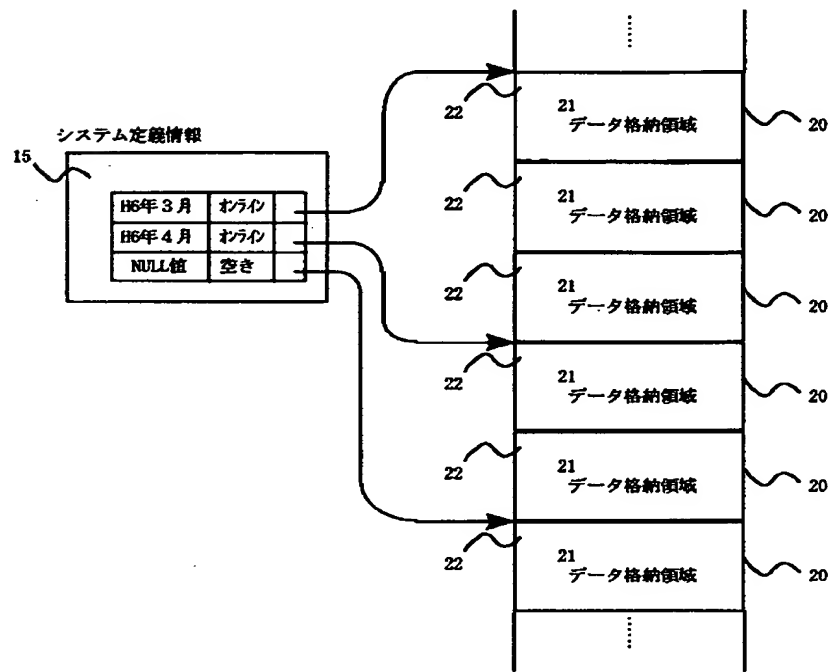
【図11】

図11



【図 12】

図 12.



【書類名】 要約書

【要約】

【課題】

大規模なデータベースにおいて、従来のインデクスやバランスドトリー方式では、検索

処理や削除処理に時間がかかり過ぎる。

【解決手段】

データベースのエリア管理の最小単位であるセグメント毎に分け、セグメント間にブックマークを設ける。データの追加時には、追加した時点の時間情報を有したブックマーク情報を作成して追加した一連のセグメント群を管理する。特定のブックマーク以降のデータをまとめて削除する場合、セグメント単位に空エリア化する。

【効果】

時系列順に到着するデータの記憶及び削除以外のデータ変更がない検索用データベースにおいて、高速に検索しオンライン中であってもデータロード・削除処理を実現することができる。

【選択図】 図2

【書類名】
【訂正書類】

職権訂正データ
特許願

<認定情報・付加情報>

【特許出願人】

【識別番号】

000005108

【住所又は居所】

東京都千代田区神田駿河台四丁目6番地

【氏名又は名称】

株式会社日立製作所

【代理人】

申請人

【識別番号】

100068504

【住所又は居所】

東京都千代田区丸の内1-5-1 株式会社日立製作所 知的所有権本部内

【氏名又は名称】

小川 勝男

出 願 人 履 歴 情 報

識別番号 [000005108]

1. 変更年月日 1990年 8月31日

[変更理由] 新規登録

住 所 東京都千代田区神田駿河台4丁目6番地
氏 名 株式会社日立製作所